BAB IV

IMPLEMENTASI

4.1 Implementasi

Pada bab implementasi akan dilakukan pengujian terhadap sistem yang baru. Sistem dianalisis dan didesain secara terperinci telah selesai maka langkah selanjutnya adalah menuju tahap implementasi. Tujuan implementasi adalah untuk menjelaskan tentang manual modul kepada semua *user* yang akan menggunakan sistem. Sehingga *user* tersebut dapat merespon apa yang ditampilkan di sistem dan memberikan masukan kepada pembuat sistem untuk dilakukan perbaikan agar sistem lebih baik lagi.

4.2 Lingkungan Implementasi

Untuk mendukung aplikasi yang akan diterapkan pada lingkungan implementasi, maka dalam hal ini harus memperhatikan perangkat keras dan perangkat lunak yang menunjang dalam Pembangunan Aplikasi E-commerce pada toko Rumah Buku Bandung.

4.2.1 Implementasi Perangkat Keras

Spesifikasi perangkat keras minimal yang dibutuhkan untuk mengimplementasikan aplikasi yang dibangun dibutuhkan spesifikasi perangkat keras antara lain terdapat pada tabel 4.1 sebagai berikut.

Tabel 4.1 Perangkat Keras yang dibutuhkan

No	Perangkat Keras	Spesifikasi
1	Prosessor	Intel Pentium 4 2.8 Ghz
2	Monitor	Monitor VGA 15”
3	VGA	Onboard 64 Mb

4	Memori	DDR2 1GB
5	Hard Disk	120 GB
6	Keyboard	Keyboard
7	Mouse	Mouse
8	Printer	Hitam putih dan warna
9	Jaringan Internet	Ada

4.2.2 Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang dibutuhkan untuk mengimplementasikan *website e-commerce* yang dibangun antara lain terdapat pada tabel 4.2 sebagai berikut.

Tabel 4.2Perangkat Lunak yang digunakan

No	Perangkat Lunak	Spesifikasi
1	Sistem operasi	Windows Xp, Mac Os X
2	Web Browser	Mozilla firefox, safari, internet explorer, Google Chrome

4.3 Implementasi Basis Data

Implementasi data dalam aplikasi *e-commerce* pada toko Rumah Buku ini menggunakan satu buah DBMS (Data *Base Management System*) dan sebuah GUI *database* untuk memudahkan tugas *administrator*. DBMS yang digunakan untuk menimplentasikan datanya adalah MySQL. Rancangan data beserta nama file query yang dilakukan adalah sebagai berikut:

1. Tabel user_grup

   Tabel user_grup berfungsi untuk menampung data grup user. Adapun *query* yang digunakan dapat dilihat pada tabel 4.3.

Tabel 4.3 Struktur tabel user_grup

Query SQL
`CREATE TABLE IF NOT EXISTS `user_grup` ( `id_grup` int(2) NOT NULL auto_increment, `nama_grup` varchar(20) NOT NULL, PRIMARY KEY (`id_grup`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4;`

2. Tabel buku

Tabel buku berfungsi untuk menampung data buku. *Query* yang digunakan untuk membangun tabel buku dapat dilihat pada tabel 4.4

Tabel 4.4 Struktur tabel buku

Query SQL

```
CREATE TABLE IF NOT EXISTS `buku` (
  `id_kategori` int(3) NOT NULL,
  `isbn` varchar(17) collate utf8_bin NOT NULL,
  `judul` varchar(128) collate utf8_bin NOT NULL,
  `pengarang` varchar(32) collate utf8_bin NOT NULL,
  `deskripsi` text collate utf8_bin NOT NULL,
  `quantity` int(4) NOT NULL default '0',
  `gambar` varchar(255) collate utf8_bin default NULL,
  `id_penerbit` int(3) NOT NULL,
  `harga` int(11) NOT NULL,
  `diskon` float(3,2) NOT NULL,
  `berat` decimal(5,2) NOT NULL default '0.00',
  `Halaman` int(4) NOT NULL,
  `ukuran` varchar(10) collate utf8_bin NOT NULL,
  `tgl_masuk` datetime NOT NULL default '0000-00-00 00:00:00',
  `tgl_ubah` datetime NOT NULL default '0000-00-00 00:00:00',
  `id_sub_kategori` int(3) NOT NULL,
  `dihapus` varchar(1) collate utf8_bin NOT NULL,
  PRIMARY KEY  (`isbn`),
  KEY `id_kategori` (`id_kategori`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
ALTER TABLE `buku`
  ADD CONSTRAINT `buku_ibfk_3` FOREIGN KEY (`id_sub_kategori`)
REFERENCES `kategori_sub` (`id_sub_kategori`),
  ADD CONSTRAINT `buku_ibfk_1` FOREIGN KEY (`id_kategori`)
REFERENCES `kategori` (`id_kategori`),
  ADD CONSTRAINT `buku_ibfk_2` FOREIGN KEY (`id_penerbit`)
REFERENCES `penerbit` (`id_penerbit`);
```

3. Tabel kabkota

Tabel kabkota digunakan untuk menyimpan data kabupaten dan kota. *Query* yang digunakan untuk membangun tabel kabkota dapat dilihat pada tabel 4.5

Tabel 4.5 Struktur tabel kabkota

Query SQL

```
CREATE TABLE IF NOT EXISTS `kabkota` (
  `id_kabkota` int(4) NOT NULL,
  `id_prov` int(3) NOT NULL,
  `nama` varchar(50) NOT NULL,
  `ongkir` int(11) NOT NULL,
  PRIMARY KEY  (`id_kabkota`), KEY `id_prov` (`id_prov`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
ALTER TABLE `kabkota`
  ADD CONSTRAINT `kabkota_ibfk_1` FOREIGN KEY (`id_prov`)
REFERENCES `provinsi` (`id_prov`);
```

4. Tabel kategori

Tabel kategori digunakan untuk menyimpan semua data kategori buku. *Query* yang digunakan untuk membangun tabel kategori dapat dilihat pada tabel 4.6

Tabel 4.6 Struktur tabel kategori

Query SQL
CREATE TABLE IF NOT EXISTS `kategori` (`id_kategori` int(3) NOT NULL auto_increment, `parent` varchar(20) collate utf8_bin NOT NULL, `nama` varchar(32) collate utf8_bin NOT NULL, `dihapus` varchar(1) collate utf8_bin NOT NULL default 'N', PRIMARY KEY (`id_kategori`), KEY `name` (`nama`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=11 ;

5. Tabel pemesanan

Tabel pemesanan digunakan untuk menyimpan data pemesanan. *Query* yang digunakan untuk membangun tabel pemesanan dapat dilihat pada tabel 4.7

Tabel 4.7 Struktur tabel pemesanan

Query SQL
CREATE TABLE IF NOT EXISTS `pemesanan` (`id_pemesanan` int(5) NOT NULL auto_increment, `id_user ` int(2) NOT NULL, `nama` varchar(20) collate utf8_bin NOT NULL, `telepon` varchar(13) collate utf8_bin NOT NULL, `email` varchar(32) collate utf8_bin NOT NULL, `alamat` varchar(128) collate utf8_bin NOT NULL, `kodepos` varchar(5) collate utf8_bin NOT NULL, `id_kota` int(4) NOT NULL,`id_provinsi` int(3) NOT NULL, `tgl_pesan` date NOT NULL, `jenis_pembayaran` varchar(15) collate utf8_bin NOT NULL, `status_pembayaran` varchar(20) collate utf8_bin NOT NULL, `status_pengiriman` varchar(20) collate utf8_bin NOT NULL, `nama_pengiriman` varchar(32) collate utf8_bin NOT NULL, `telepon_pengiriman` varchar(13) collate utf8_bin NOT NULL, `email_pengiriman` varchar(32) collate utf8_bin NOT NULL, `alamat_pengiriman` varchar(128) collate utf8_bin NOT NULL, `kodepos_pengiriman` varchar(5) collate utf8_bin NOT NULL, `id_kota_pengiriman` int(4) NOT NULL,`id_provinsi_pengiriman` int(3) NOT NULL, `kode_konfirmasi` varchar(12) NOT NULL, `total` int(11) NOT NULL, `berat` int(11) NOT NULL, `no_resi` varchar(13) collate utf8_bin NOT NULL, PRIMARY KEY (`id_pemesanan`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=51 ; ALTER TABLE `pemesanan` ADD CONSTRAINT `pemesanan_ibfk_2` FOREIGN KEY (`id_prov`) REFERENCES `provinsi` (`id_prov`),

```
        ADD CONSTRAINT `pemesanan_ibfk_1` FOREIGN KEY (`id_kabkota`)
REFERENCES `kabkota` (`id_kabkota`);
CONSTRAINT `pemesanan_ibfk_3`FOREIGN KEY (`id_user` ) REFERENCES `user`
(`id_user` )
```

6. Tabel pemesanan_detail

   Tabel pemesanan_detail digunakan untuk menyimpan data pemesanan detail. *Query* yang digunakan untuk membangun tabel pemesanan detail dapat dilihat pada tabel 4.8

Tabel 4.8 Struktur tabel pemesanan_detail

Query SQL

```
      CREATE TABLE IF NOT EXISTS `pemesanan_detail` (
        `id_pemesanan_detail` int(5) NOT NULL auto_increment,
        `id_pemesanan` int(5) NOT NULL,
        `judul` varchar(128) collate utf8_bin NOT NULL,
        `isbn` varchar(17) collate utf8_bin NOT NULL,
        `harga` int(11) NOT NULL,
        `quantity` int(4) NOT NULL default '0',
        PRIMARY KEY  (`id_pemesanan_detail`),
        KEY `id_pemesanan` (`id_pemesanan`)
      ) ENGINE=InnoDB  DEFAULT CHARSET=utf8 COLLATE=utf8_bin
AUTO_INCREMENT=68 ;
      ALTER TABLE `pemesanan_detail`
      ADD    CONSTRAINT    `pemesanan_detail_ibfk_2`    FOREIGN    KEY
(`id_pemesanan`) REFERENCES `pemesanan` (`id_pemesanan`),
ADD CONSTRAINT `pemesanan_detail_ibfk_1` FOREIGN KEY (`isbn`) REFERENCES
`buku` (`isbn`);
```

7. Tabel penerbit

   Tabel penerbit digunakan untuk menyimpan data penerbit. *Query* yang digunakan untuk membangun tabel penerbit dapat dilihat pada tabel 4.9

Tabel 4.9 Struktur tabel penerbit

Query SQL

```
      CREATE TABLE IF NOT EXISTS `penerbit` (
        `id_penerbit` int(3) NOT NULL auto_increment,
        `nama_penerbit` varchar(32) collate utf8_bin NOT NULL,
        `alamat_penerbit` varchar(128) collate utf8_bin NOT NULL,
        `dihapus` varchar(1) collate utf8_bin NOT NULL default 'N',
        PRIMARY KEY  (`id_penerbit`)
      ) ENGINE=InnoDB  DEFAULT CHARSET=utf8 COLLATE=utf8_bin
AUTO_INCREMENT=8;
```

8. Tabel provinsi

   Tabel provinsi digunakan untuk menyimpan data provinsi. *Query* yang digunakan untuk membangun tabel provinsi dapat dilihat pada tabel 4.10

Tabel 4.10 Struktur tabel provinsi

Query SQL
CREATE TABLE IF NOT EXISTS `provinsi` (`id_prov` int(3) NOT NULL, `nama` varchar(100) NOT NULL, PRIMARY KEY (`id_prov`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

9. Tabel *contact*

   Tabel *contact* digunakan untuk menyimpan data pertanyaan dari halaman kontak kami. *Query* yang digunakan untuk membangun tabel *contact* dapat dilihat pada tabel 4.11

Tabel 4.11 Struktur tabel *contact*

Query SQL
CREATE TABLE IF NOT EXISTS `contact` (`id_kontak` int(5) NOT NULL auto_increment, `nama` varchar(32) NOT NULL, `alamat` varchar(128) NOT NULL, `email` varchar(32) NOT NULL, `telepon` varchar(13) NOT NULL, `pesan` text NOT NULL, `tanggal` date NOT NULL, `tgl_balas` date NOT NULL, `pesan_balas` text NOT NULL, PRIMARY KEY (`id_kontak`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=15 ;

10. Tabel kategori_sub

    Tabel kategori_sub digunakan untuk menyimpan data sub kategori. *Query* yang digunakan untuk membangun tabel kategori_sub dapat dilihat pada tabel 4.12

Tabel 4.12 Struktur tabel kategori_sub

Query SQL
CREATE TABLE IF NOT EXISTS `kategori_sub` (`id_sub_kategori` int(3) NOT NULL auto_increment, `nama_subkategori` varchar(32) collate utf8_bin NOT NULL default '0', `id_kategori` int(3) NOT NULL default '0', `dihapus` varchar(1) collate utf8_bin NOT NULL, PRIMARY KEY (`id_sub_kategori`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=27 ; ALTER TABLE `kategori_sub` ADD CONSTRAINT `kategori_sub_ibfk_1` FOREIGN KEY (`id_kategori`) REFERENCES `kategori` (`id_kategori`);

11. Tabel konfirmasi

Tabel konfirmasi digunakan untuk menyimpan data dari hasil konfirmasi pembayaran yang dilakukan oleh. *Query* yang digunakan untuk membangun tabel konfirmasi dapat dilihat pada tabel 4.13

Tabel 4.13 Struktur tabel konfirmasi

Query SQL
CREATE TABLE IF NOT EXISTS `konfirmasi` (`id_konfirmasi` int(5) NOT NULL auto_increment, `nama` varchar(32) NOT NULL, `email` varchar(32) NOT NULL, `id_pemesanan ` int(5) NOT NULL, `jumlah` int(11) NOT NULL, `bank` varchar(32) NOT NULL, `bank_konsumen` varchar(32) NOT NULL, `nama_pengirim` varchar(32) NOT NULL, `tgl_bayar` date NOT NULL, `dicek` varchar(1) NOT NULL, PRIMARY KEY (`id_konfirmasi`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=35 ; ALTER TABLE `konfirmasi` ADD CONSTRAINT `konfirmasi_ibfk_1` FOREIGN KEY (`id_pemesanan`) REFERENCES `pemesanan` (`id_pemesanan`);

12. Tabel *User*

Tabel user_grup digunakan untuk menyimpan data dari user_grup. *Query* yang digunakan untuk membangun tabel user_grup dapat dilihat pada tabel 4.14

Tabel 4.14 Struktur tabel *user*

Query SQL

```
CREATE TABLE IF NOT EXISTS `user` (
  `id_user` int(2) NOT NULL AUTO_INCREMENT,
  `id_grup` int(2) NOT NULL,
  `username` varchar(20) COLLATE utf8_bin NOT NULL,
  `password` varchar(51) COLLATE utf8_bin NOT NULL,
  `aktif` varchar(1) COLLATE utf8_bin NOT NULL,
  PRIMARY KEY (`id_user`),
  KEY `id_grup` (`id_grup`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin
AUTO_INCREMENT=4 ;
ALTER TABLE `user`
  ADD CONSTRAINT `user_ibfk_1` FOREIGN KEY (`id_grup`) REFERENCES
`user_grup` (`id_grup`);
```

13. Tabel V_Konsumen

Tabel V_konsumen digunakan untuk menyimpan data konsumen yang di dapat dari tabel pemesanan. *Query* yang digunakan untuk membangun tabel v_konsumen dapat dilihat pada tabel 4.15

Tabel 4.15 Struktur tabel v_konsumen

Query SQL

```
CREATE VIEW `rumahbuk_ci_projek`.`v_konsumen` AS select
count(`p`.`nama`) AS `jml_memesan`,`p`.`nama` AS `nama`,
`p`.`email` AS `email`,`p`.`telepon` AS `telepon`,
`p`.`alamat` AS `alamat`,
`p`.`kodepos` AS `kodepos`,`k`.`nama` AS `kota`,
`r`.`nama` AS `provinsi` from
((`rumahbuk_ci_projek`.`pemesanan` `p` join
`rumahbuk_ci_projek`.`kabkota` `k`
on((`p`.`id_kota` = `k`.`id_kabkota`)))
join `rumahbuk_ci_projek`.`provinsi` `r`
on((`p`.`id_provinsi` = `r`.`id_prov`)))
group by `p`.`nama`,`p`.`email` order by count(`p`.`nama`) desc;
```

14. Tabel *Order*

Tabel *order* digunakan untuk menyimpan data *order* buku dimana konsumen akan memesan lebih dari jumlah stok buku yang ada. *Query* yang digunakan untuk membangun tabel *order* dapat dilihat pada tabel 4.16

Tabel 4.16 Struktur tabel *order*

Query SQL

```
CREATE TABLE IF NOT EXISTS `order` (
  `id_order` int(5) NOT NULL auto_increment,
  `tgl_order` date NOT NULL,
  `email` varchar(32) NOT NULL,
  `telepon` varchar(13) NOT NULL,
  `status` varchar(1) NOT NULL,
  PRIMARY KEY  (`id_order`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;
```

15. Tabel *Order_detail*

Tabel *order_detail* digunakan untuk menyimpan data order buku dimana konsumen akan memesan lebih dari jumlah stok buku yang ada. *Query* yang digunakan untuk membangun tabel *order_detail* dapat dilihat pada tabel 4.17

Tabel 4.17 Struktur tabel *order_detail*

Hasil

```
CREATE TABLE IF NOT EXISTS `order_detail` (
  `id_order_detail` int(11) NOT NULL auto_increment,
  `id_order` int(5) NOT NULL,
  `isbn` varchar(17) NOT NULL,
  `jml` int(5) NOT NULL,
  PRIMARY KEY  (`id_order_detail`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=9 ;
ALTER TABLE `order_detail`
  ADD  CONSTRAINT  `order_detail_ibfk_2`  FOREIGN  KEY  (`isbn`)
REFERENCES `buku` (`isbn`),
  ADD CONSTRAINT `order_detail_ibfk_1` FOREIGN KEY (`id_order`)
REFERENCES `order` (`id_order`);
```

16. Tabel pengadaan

Tabel pengadaan digunakan untuk menyimpan data pengadaan buku. *Query* yang digunakan untuk membangun tabel *order_detail* dapat dilihat pada tabel 4.18

Tabel 4.18 Struktur tabel pengadaan

Query SQL
CREATE TABLE IF NOT EXISTS `pengadaan` (`id_pengadaan` int(5) NOT NULL auto_increment, `id_penerbit` int(3) NOT NULL, `id_user` int(2) NOT NULL, `tgl_pengadaan` date NOT NULL, `diproses` varchar(1) NOT NULL, PRIMARY KEY (`id_pengadaan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ; ALTER TABLE `pengadaan` ADD CONSTRAINT `pengadaan_ibfk_1` FOREIGN KEY (`id_penerbit`) REFERENCES `penerbit` (`id_penerbit`);ADD CONSTRAINT `pengadaan_ibfk_2` FOREIGN KEY (`id_user`) REFERENCES `user` (`id_user`)

17. Tabel v_pemesanan

Tabel v_pemesanan adalah tabel *view* dari pemesanan yang digunakan untuk menyimpan data pemesanan. *Query* yang digunakan untuk membangun tabel v_pemesanan dapat dilihat pada tabel 4.19

Tabel 4.19 Struktur tabel v_pemesanan

Query SQL
CREATE VIEW `rumahbuk_ci_projek`.`v_pemesanan` AS select `p`.`kode_konfirmasi` AS `kode_konfirmasi`,`p`.`id_pemesanan` AS `id_pemesanan`,`p`.`nama` AS `nama`,`p`.`telepon` AS `telepon`,`p`.`email` AS `email`,`p`.`alamat` AS `alamat`,`p`.`kodepos` AS `kodepos`,`k1`.`nama` AS `kota`,`p1`.`nama` AS `provinsi`,date_format(`p`.`tgl_pesan`,_utf8'%d-%m-%Y') AS `tgl_pesan`,`p`.`status_pembayaran` AS `status_pembayaran`,`p`.`jenis_pembayaran` AS `jenis_pembayaran`,`p`.`status_pengiriman` AS `status_pengiriman`,`p`.`nama_pengiriman` AS `nama_pengiriman`,`p`.`telepon_pengiriman` AS `telepon_pengiriman`,`p`.`email_pengiriman` AS `email_pengiriman`,`p`.`alamat_pengiriman` AS `alamat_pengiriman`,`p`.`kodepos_pengiriman` AS `kodepos_pengiriman`,`k2`.`nama` AS `kota_pengiriman`,`p2`.`nama` AS `provinsi_pengiriman`,`p`.`total` AS `total`,`p`.`berat` AS `berat`,`p`.`no_resi` AS `no_resi`,(`k2`.`ongkir` * `p`.`berat`) AS `ongkir` from ((((`rumahbuk_ci_projek`.`pemesanan` `p` join `rumahbuk_ci_projek`.`kabkota` `k1` on((`p`.`id_kota` = `k1`.`id_kabkota`))) join `rumahbuk_ci_projek`.`provinsi` `p1` on((`p`.`id_provinsi` = `p1`.`id_prov`))) join `rumahbuk_ci_projek`.`kabkota` `k2` on((`p`.`id_kota_pengiriman` = `k2`.`id_kabkota`))) join `rumahbuk_ci_projek`.`provinsi` `p2` on((`p`.`id_provinsi_pengiriman` = `p2`.`id_prov`)));

18. Tabel pengadaan detail

Tabel pengadaan detail digunakan untuk menyimpan detail dari data pengadaan buku. *Query* yang digunakan untuk membangun tabel pengadaan_*detail* dapat dilihat pada tabel 4.20

Tabel 4.20 Struktur tabel pengadaan detail

Query SQL

```
CREATE TABLE IF NOT EXISTS `pengadaan_detail` (
  `id_pengadaan_detail` int(5) NOT NULL auto_increment,
  `id_pengadaan` int(5) NOT NULL,
  `isbn` varchar(17) NOT NULL,
  `jml` int(4) NOT NULL,
  PRIMARY KEY  (`id_pengadaan_detail`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=24 ;
ALTER TABLE `pengadaan_detail`
  ADD CONSTRAINT `pengadaan_detail_ibfk_2` FOREIGN KEY (`isbn`)
REFERENCES `buku` (`isbn`),
  ADD CONSTRAINT `pengadaan_detail_ibfk_1` FOREIGN KEY
(`id_pengadaan`) REFERENCES `pengadaan` (`id_pengadaan`);
```

19. Tabel retur

Tabel retur digunakan untuk menyimpan data retur buku. *Query* yang digunakan untuk membangun tabel retur dapat dilihat pada tabel 4.21

Tabel 4.21 Struktur tabel retur

Query SQL

```
CREATE TABLE IF NOT EXISTS `retur` (
  `id_retur` int(3) NOT NULL auto_increment,
  `id_pemesanan` int(5) NOT NULL,
`id_user` int(2) NOT NULL,
  `no_resi` varchar(13) character set utf8 collate utf8_bin NOT
NULL,
  `alasan` text NOT NULL,
  `tanggal_retur` date NOT NULL,
  `status` varchar(20) NOT NULL,
`no_resi_retur` varchar(13) character set utf8 collate utf8_bin
NOT NULL,
`tgl_penerimaan` date NOT NULL,
`nama_penerima ` varchar(20) character set utf8 collate utf8_bin
NOT NULL,
`waktu_diterima ` varchar(6) character set utf8 collate utf8_bin
NOT NULL,
  PRIMARY KEY  (`id_retur`),
  KEY `no_resi` (`no_resi`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=9 ;
ALTER TABLE `retur`
  ADD CONSTRAINT `retur_ibfk_1` FOREIGN KEY (`id_pemesanan`)
REFERENCES `pemesanan` (`id_pemesanan`); ADD CONSTRAINT `retur_ibfk_2`
FOREIGN KEY (`id_user`) REFERENCES `user` (`id_user`);
```

20. Tabel retur detail

Tabel retur digunakan untuk menyimpan detail dari data retur buku. *Query* yang digunakan untuk membangun tabel *order_detail* dapat dilihat pada tabel 4.22

Tabel 4.22 Struktur tabel retur detail

Query SQL
CREATE TABLE IF NOT EXISTS `retur_detail` (`id_retur_detail` int(3) NOT NULL auto_increment, `id_retur` int(3) NOT NULL, `judul` varchar(255) NOT NULL, `quantity` int(4) NOT NULL, PRIMARY KEY (`id_retur_detail`), KEY `id_retur` (`id_retur`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ; ALTER TABLE `retur_detail` ADD CONSTRAINT `retur_detail_ibfk_1` FOREIGN KEY (`id_retur`) REFERENCES `retur` (`id_retur`);

21. Tabel penerimaan

Tabel retur digunakan untuk menyimpan detail dari data penerimaan. *Query* yang digunakan untuk membangun tabel penerimaan dapat dilihat pada tabel 4.23

Tabel 4.23 Struktur tabel penerimaan

Query SQL
CREATE TABLE IF NOT EXISTS `penerimaan` (`id_penerimaan` int(5) NOT NULL auto_increment, `id_pemesanan` int(5) NOT NULL, `tgl_penerimaan` date NOT NULL, `nama_penerima` varchar(20) NOT NULL, `waktu_diterima` varchar(6) NOT NULL, PRIMARY KEY (`id_penerimaan`), KEY `id_pemesanan` (`id_pemesanan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=50 ;ALTER TABLE `penerimaan` ADD CONSTRAINT `penerimaan_ibfk_1` FOREIGN KEY (`id_pemesanan`) REFERENCES `pemesanan` (`id_pemesanan`);

4.4 Implementasi Class

Class-class yang terbentuk dalam *class* diagram selanjutnya akan diimplementasikan dalam bentuk file-file *.php yang nantinya file-file *.php tersebut digunakan dalam proses yang ada dalam aplikasi. Berikut implementasi *class* yang telah terdefinisi dan digunakan dalam aplikasi. Implementasi Class

pada aplikasi *E-Commerce* pada toko Rumah Buku Bandung dapat dilihat pada tabel 4.24

Tabel 4.24 Implementasi *Class*

No	Nama Class	Nama File Fisik
1	AdminController	Admin.php
2	KatalogController	Katalog.php
3	AdminModel	Admin_Model.php
4	KatalogModel	Katalog_Model.php
5	BukuView	tabel_buku.php
6	KategoriView	tabel_kategori.php
7	PenerbitView	tabel_penerbit.php
8	PemensananView	tabel_pemesanan.php
9	LaporanView	buku_terjual.php
10	UserView	tabel_user.php
11	SubkategoriView	tabel_subkategori.php
12	KontakView	lihat_kontak.php
13	PromosiView	tabel_konsumen.php
14	ReturView (admin)	tabel_retur.php
15	KeranjangView	Cart_view.php
16	BerandaView	index.php
17	ReturView (katalog)	retur.php
18	KonfirmasiView	confirm.php
19	KontakView	contact.php

4.5 Implementasi Antarmuka

Implementasi antar muka dilakukan dengan setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk file program. Berikut ini adalah implementasi antarmuka untuk pengguna. Implementasi antarmuka pada aplikasi *e-ecommerce* pada toko Rumah Buku Bandung dibagi menjadi dua bagian yaitu :

Tabel 4.25 Implementasi Antarmuka Admin

Menu	Deskripsi	Nama File
Login Admin	Digunakan sebagai halaman login *admin*	index.php/admin
Halaman Utama Admin	Halaman utama ketika *admin* berhasil login berisi lima pemesanan terakhir dan pembayaran terakhr.	index.php/admin/beranda
Data Kategori	Halaman *admin* yang digunakan untuk melihat, menambah, mengubah dan menghapus data kategori buku.	index.php/admin/datakategori

Data Sub Kategori	Halaman *admin* yang digunakan untuk melihat, menambah, mengubah dan menghapus data sub kategori.	index.php/admin/dataSubkategori
Data Penerbit	Halaman *admin* yang digunakan untuk melihat, menambah, mengubah dan menghapus data penerbit.	index.php/admin/datapenerbit
Data Buku	Halaman *admin* yang digunakan untuk melihat, menambah, mengubah dan menghapus data buku.	index.php/admin/databuku
Data Pengadaan	Halaman *admin* yang digunakan untuk melihat dan melakukan proses pengadaan buku dan mencetak surat pemesanan	index.php/admin/datapengadaan
Data Pemesanan	Halaman yang digunakan *admin* untuk mengolah data pemesanan seperti pengubahan status pemayaran dan pengiriman serta cetak label	index.php/admin/datapesan
Data Konsumen	Halaman yang digunakan *admin* untuk melihat data konsumen	index.php/admin/datakonsumen
Kirim Promosi	Halaman yang digunakan *admin* untuk melihat data konsumen dan mengirimkan *email* promosi kepada konsumen	index.php/admin/datakonsumen
Retur	Halaman yang digunakan *admin* melihat retur yang dilakukan konsumen	index.php/admin/dataretur
Balas Kontak	Halaman yang digunakan *admin* untuk meliat pesan yang dikirimkan konsumen dan membalas *email* tersebut	index.php/admin/datakontak
Laporan Penjualan	Halaman yang digunakan admin untuk melihat buku terjual dan mencetak laporan dalam periode tertentu	index.php/admin/buku_terjual
Pengaturan *User*	Halaman yang digunakan *admin* untuk melihat *user* yang terdaftar pada sistem serta menambah, mengubah dan menghapus data *user*	index.php/admin/datauser
Grup *User*	Halaman yang digunakan *admin* untuk melihat grup *user* yang terdaftar pada sistem serta menambah, mengubah dan menghapus data grup *user*	index.php/admin /datausergrup
Backup	Halaman yang digunakan admin untuk menyalin semua tabel yang ada pada database sistem	index.php/admin /backup
Akun Saya	Halaman yang digunakan untuk mengatur *username* dan *password* user	index.php/admin /akun_saya

Logout	Halaman yang digunakan untuk menghapus session admin	index.php/admin/destroy

Tabel dari implementasi dari petugas dapat dilihat pada tabel 4.26

Tabel 4.26 Implementasi Petugas

Menu	Deskripsi	Nama File
Login Petugas	Digunakan sebagai halaman login petugas	index.php/admin
Halaman Utama Petugas	Halaman utama ketika petugas berhasil login berisi lima pemesanan terakhir dan pembayaran terakhr.	index.php/admin/beranda
Data Kategori	Halaman petugas yang digunakan untuk melihat, menambah, mengubah dan menghapus data kategori buku.	index.php/admin/datakategori
Data Sub Kategori	Halaman petugas yang digunakan untuk melihat, menambah, mengubah dan menghapus data sub kategori.	index.php/admin/dataSubkategori
Data Penerbit	Halaman petugas yang digunakan untuk melihat, menambah, mengubah dan menghapus data penerbit.	index.php/admin/datapenerbit
Data Buku	Halaman petugas yang digunakan untuk melihat, menambah, mengubah dan menghapus data buku.	index.php/admin/databuku
Data Pengadaan	Halaman petugas yang digunakan untuk melihat dan melakukan proses pengadaan buku dan mencetak surat pemesanan	index.php/admin/datapengadaan
Data Pemesanan	Halaman yang digunakan petugas untuk mengolah data pemesanan seperti pengubahan status pemayaran dan pengiriman serta cetak label	index.php/admin/datapesan
Data Konsumen	Halaman yang digunakan petugas untuk melihat data konsumen	index.php/admin/datakonsumen
Kirim Promosi	Halaman yang digunakan petugas untuk melihat data konsumen dan mengirimkan *email* promosi kepada konsumen	index.php/admin/datakonsumen
Retur	Halaman yang digunakan petugas melihat retur yang dilakukan konsumen	index.php/admin/dataretur
Balas Kontak	Halaman yang digunakan petugas untuk meliat pesan yang dikirimkan konsumen dan membalas *email* tersebut	index.php/admin/datakontak
Laporan Penjualan	Halaman yang digunakan petugas untuk melihat buku terjual dan mencetak laporan dalam periode	index.php/admin/buku_terjual

	tertentu	
Akun Saya	Halaman yang digunakan untuk mengatur *username* dan *password*	index.php/admin /akun_saya
Logout	Halaman yang digunakan untuk menghapus session petugas	index.php/admin/destroy

Tabel dari implementasi dari konsumen dapat dilihat pada tabel 4.27

Tabel 4.27 Implementasi Konsumen

Menu	Deskripsi	Nama File
Beranda	Digunakan sebagai halaman utama konsumen berisi informasi mengenai buku terbaru, buku terlaris, kategori buku dan konsumen dapat melakukan tracking JNE untuk mengecek keberadaan barang yang sedang dikirim.	index.php/katalog
Tentang Kami	Digunakan untuk menampilkan informasi mengenai deskripsi Toko Rumah Buku Bandung	index.php/katalog/about
Keranjang Belanja	Digunakan untuk melihat, merubah, menghapus barang dan detail belanjaan yang akan dilakukan	index.php/katalog/show_cart
Kontak Kami	Halaman yang digunakan untuk memberikan informasi mengenai alamat, no.telp/fax, email dan form yang dapat diisi oleh konsumen apabila ada pesan saran ataupun kritik mengenai toko rumah buku Bandung	index.php/katalog/contact
Bantuan	Halaman yang digunakan untuk memberikan informasi mengenai panduan cara berbelanja dan cara pembayaran melalui *e-commerce* toko rumah buku Bandung	index.php/katalog/help
Retur Barang	Halaman yang digunakan konsumen untuk melakukan retur barang apabila barang yang diterima oleh konsumen terdapat cacat dalam kondisi buku	index.php/katalog/retur.php
Konfirmasi Pembayaran	Halaman yang digunakan untuk melakukan konfirmasi pembayaran apabila konsumen telah melakukan pembayaran melalui Bank yang dipilih oleh konsumen	index.php/katalog/confirm

4.6 Pengujian Perangkat Lunak

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan-kesalahan atau kekurangan-kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

Pengujian yang digunakan untuk menguji sistem yang baru adalah metode pengujian *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.

4.6.1 Rencana Pengujian

Rencana pengujian yang akan dilakukan dengan menguji sistem secara *alpha* dan *beta*. Pengujian pembangunan aplikasi *e-commerce* pada Toko Rumah Buku Bandung menggunakan data uji berdasarkan data dari Toko Rumah Buku. Rencana pengujian selengkapnya terlihat pada tabel 4.28

Tabel 4.28 Rencana Pengujian Aplikasi Admin

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Isi data Login	*Black Box*
	Validasi username and password	*Black Box*
Pengolahan Data Kategori	Tambah Kategori	*Black Box*
	Ubah Kategori	*Black Box*
	Hapus Kategori	*Black Box*
	Cari Kategori	*Black Box*
Pengolahan Data Sub Kategori	Tambah SubKategori	*Black Box*
	Ubah SubKategori	*Black Box*
	Hapus SubKategori	*Black Box*
	Cari Subkategori	*Black Box*
Pengolahan Data Penerbit	Tambah Penerbit	*Black Box*
	Ubah Penerbit	*Black Box*
	Hapus Penerbit	*Black Box*
	Cari Penerbit	*Black Box*
Pengolahan Data Buku	Tambah Buku	*Black Box*
	Ubah Buku	*Black Box*
	Hapus Buku	*Black Box*
	Cari Buku	*Black Box*
Pengadaan	Cetak Pemesanan	*Black Box*
	Pengadaan stok buku	*Black Box*
Pengolahan Data Pemesanan	Ubah Pemesanan	*Black Box*
	Konfirmasi Pembayaran	*Black Box*
Kirim Promosi	Kirim email promosi	*Black Box*
Retur	Ubah status retur	*Black Box*
Balas Kontak	Balas kontak email	*Black Box*
Pengaturan User	Tambah user	*Black Box*
	Ubah User	*Black Box*
	Hapus User	*Black Box*
Grup User	Tambah grup user	*Black Box*
	Ubah grup user	*Black Box*
Laporan Penjualan	Cetak laporan penjualan	*Black Box*
Akun Saya	Ubah akun	*Black Box*

Logout	Keluar sistem	*Black Box*

Rencana pengujian dari aplikasi konsumen dapat dilihat pada tabel 4.29

Tabel 4.29 Rencana Pengujian Aplikasi Konsumen

Kelas Uji	Butir Uji	Jenis Pengujian
Keranjang Belanja	Tambah data barang ke keranjang	*Black Box*
	Update keranjang	*Black Box*
	Hapus data barang di keranjang	*Black Box*
	Masukan data pemesanan	*Black Box*
	Pesan buku lebih dari stok	*Black Box*
Kontak Kami	Tampil informasi mengenai nama, alamat, no.telp, email dan form yang dapat diisi oleh konsumen apabila ada pesan, saran ataupun kritik mengenai toko rumah buku Bandung	*Black Box*
Konfirmasi Pembayaran	*Input* konfirmasi pembayaran Bank	*Black Box*
Retur	Halaman yang digunakan konsumen untuk melakukan retur barang apabila barang yang diterima oleh konsumen terdapat cacat dalam kondisi buku	*Black Box*
Cari Buku	cari buku	*Black Box*

4.6.2 Pengujian Alpha

Hasil implementasi perangkat lunak dari aplikasi *e-commerce* yang dibangun harus diuji terlebih dahulu agar sistem yang dibangun dapat berjalan dengan baik. Pengujian perangkat lunak sistem penerapan *e-commerce* pada toko Rumah Buku Bandung menggunakan data uji berdasarkan form yang terdapat didalam aplikasi.

4.6.2.1 Pengujian aplikasi *admin*

Pengujian yang dilakukan untuk halaman *admin* adalah menguji semua proses yang ada di halaman admin diantaranya adalah sebagai berikut :

1. Pengujian *login*

   Pengujian *login* dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian *login* dilihat pada tabel 4.30

Tabel 4.30 Pengujian login

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Username : admin *Password* : admin	Tercantum pada *text field username* dan *password*	Dapat mengisi data *login* sesuai yang diharapkan.	[√] Diterima [ ] Ditolak
Data Login berupa *Username* : admin *Password* : admin	Data login dicari dalam *database, username* dan data *password* benar login *valid* dan dapat masuk ke menu utama	*Username* dan *Password* benar dapat masuk ke menu utama	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji salah (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Username : admin *password* : 1234	Tidak dapat login/masuk dan menampilkan pesan kesalahan	Muncul pesan "*Username* atau *password* salah"	[√] Diterima [ ] Ditolak

2. Pengujian *logout*

   Pengujian *logout* dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian *logout* dilihat pada tabel 4.31

Tabel 4.31 Pengujian *logout*

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Admin yang sudah *login* memilih menu *logout*	Sistem akan menampilkan menu login admin	Sistem akan menampilkan menu login admin	[√] Diterima [ ] Ditolak

3. Pengujian tambah kategori

   Pengujian tambah kategori dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian tambah kategori dilihat pada tabel 4.32

Tabel 4.32 Pengujian tambah kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Tambah kategori			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Tambah kategori	Menampilkan *form* kategori	Sistem menampilkan *form* tambah kategori	[√] Diterima [ ] Ditolak
Mengisi data kategori contoh : Kategori : Agama	Tercantun dalam masing-masing *field* kemudian data kategori tersimpan dalam *database*.	Data kategori berhasil disimpan dalam *database*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Tambah kategori			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data kategori contoh : Kategori : Agama	Kategori tidak dapat disimpan dalam *database* karena data kategori sudah ada dan muncul pesan kesalahan	Sistem tidak dapat menyimpan data kategori dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

4. Pengujian ubah kategori

   Pengujian ubah kategori dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian ubah kategori dilihat pada tabel 4.33

Tabel 4.33 Pengujian ubah kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Ubah kategori			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Ubah kategori	Menampilkan form ubah kategori beserta data yang akan diubah	Dapat menampilkan form ubah kategori beserta data yang akan diubah	[√] Diterima [ ] Ditolak
Data kategori lama contoh : Kategori : Agama	Tercantun dalam *field* kategori berupa nama kategori.	Dapat menampilkan field data kategori yang akan diubah.	[√] Diterima [ ] Ditolak
Data kategori yang akan diubah contoh : Kategori : Semua Agama	Dapat mengisi *field* data yang akan diubah dengan benar dan data kategori berhasil diubah dan disimpan dalam *database*.	Dapat mengisi *field* data yang akan diubah dengan benar dan data kategori berhasil diubah dan disimpan dalam *databsase*	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji Salah (Data Salah)			
Ubah kategori			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data kategori yang akan diubah contoh : Kategori : Agama Islam Kristen protestan kong	Data kategori tidak berhasil diubah dan menampilkan pesan kesalahan, maxlength adalah 32	Sistem tidak dapat menyimpan data kategori dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

5. Pengujian hapus kategori

   Pengujian ubah kategori dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian hapus kategori dilihat pada tabel 4.34

Tabel 4.34 Pengujian hapus kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Hapus kategori			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data kategori yang akan dihapus : contoh : Kategori : Agama	Tercantum data kategori yang akan dihapus berupa setelah memilih hapus maka tampil pesan peringatan apakah data akan dihapus	Tercantum data kategori yang akan dihapus berupa setelah memilih hapus maka tampil pesan peringatan apakah data akan dihapus	[√] Diterima [ ] Ditolak

6. Pengujian cari kategori

   Pengujian cari kategori dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian cari kategori dilihat pada tabel 4.35

Tabel 4.35 Pengujian cari kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Cari kategori			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data kategori	Menampilkan data kategori	Berhasil menampilkan data kategori	[√] Diterima [ ] Ditolak

Data yang akan dicari berdasarkan nama kategori contoh : Agama	Menampilkan data kategori berdasarkan nama kategori yang dicari	Berhasil Menampilkan data kategori berdasarkan nama kategori yang dicari	[√] Diterima [ ] Ditolak

7. Pengujian tambah subkategori

Pengujian tambah subkategori dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian tambah subkategori dilihat pada tabel 4.36

Tabel 4.36 Pengujian tambah subkategori

Kasus dan Hasil Uji Benar (Data Benar)			
Tambah subkategori			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Tambah subkategori	Menampilkan *form* subkategori	Sistem menampilkan *form* tambah subkategori	[√] Diterima [ ] Ditolak
Mengisi data subkategori contoh : Kategori : Agama Subkategori : Agama Islam	Tercantun dalam masing-masing *field* kemudian data subkategori tersimpan dalam *database*.	Data subkategori berhasil disimpan dalam *database*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Tambah subkategori			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data subkategori contoh : Kategori : Agama Subkategori : Agama Islam	Subkategori tidak dapat disimpan dalam *database* karena data subkategori sudah ada dan muncul pesan kesalahan	Sistem tidak dapat menyimpan data kategori dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

8. Pengujian ubah subkategori

Pengujian tambah subkategori dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian tambah subkategori dilihat pada tabel 4.37

Tabel 4.37 Pengujian ubah subkategori

Kasus dan Hasil Uji Benar (Data Benar)			
Ubah subkategori			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Ubah subkategori	Menampilkan form ubah subkategori beserta data yang akan diubah	Dapat menampilkan form ubah subkategori beserta data yang akan diubah	[√] Diterima [ ] Ditolak
Data subkategori lama contoh : Kategori : Agama Subkategori : Agama Islam	Tercantun dalam *field* kategori dan subkategori.	Dapat menampilkan field data subkategori yang akan diubah.	[√] Diterima [ ] Ditolak
Data subkategori yang akan diubah contoh : Kategori : Agama Subtakategori : Agama islam	Dapat mengisi *field* data yang akan diubah dengan benar dan data subkategori berhasil diubah dan disimpan dalam *database*,	Dapat mengisi *field* data yang akan diubah dengan benar dan data subkategori berhasil diubah dan disimpan dalam *databsase*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Ubah subkategori			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data subkategori yang akan diubah contoh : Kategori : Agama Subkategori : Agama Islam Kristen protestan kong	Data subkategori tidak berhasil diubah dan menampilkan pesan kesalahan, maxlength subkategotri adalah 32	Sistem tidak dapat menyimpan data subkategori dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

9. Pengujian hapus subkategori

   Pengujian hapus subkategori dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian hapus subkategori dilihat pada tabel 4.38

Tabel 4.38 Pengujian hapus subkategori

Kasus dan Hasil Uji Benar (Data Benar)			
Hapus kategori			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data subkategori yang akan dihapus : contoh : Kategori : Agama Subkategori : Agama Islam	Tercantum data subkategori yang akan dihapus berupa setelah memilih hapus maka tampil pesan peringatan apakah data akan dihapus	Tercantum data subkategori yang akan dihapus berupa setelah memilih hapus maka tampil pesan peringatan apakah data akan dihapus	[√] Diterima [ ] Ditolak

10. Pengujian cari subkategori

Pengujian cari subkategori dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian cari subkategori dilihat pada tabel 4.39

Tabel 4.39 Pengujian cari subkategori

Kasus dan Hasil Uji Benar (Data Benar)			
Cari subkategori			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data subkategori	Menampilkan data subkategori	Berhasil menampilkan data subkategori	[√] Diterima [ ] Ditolak
Data yang akan dicari berdasarkan nama subkategori contoh : Agama Islam	Menampilkan data kategori berdasarkan nama subkategori yang dicari	Berhasil Menampilkan data kategori berdasarkan nama subkategori yang dicari	[√] Diterima [ ] Ditolak

11. Pengujian tambah penerbit

Pengujian tambah penerbit dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian tambah penerbit dilihat pada tabel 4.40

Tabel 4.40 Pengujian tambah penerbit

Kasus dan Hasil Uji Benar (Data Benar)			
Tambah penerbit			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Tambah penerbit	Menampilkan *form* tambah penerbit	Sistem berhasil menampilkan *form* tambah penerbit	[√] Diterima [ ] Ditolak
Mengisi data penerbit contoh : Penerbit : Gramedia	Tercantun dalam masing-masing *field* kemudian data penerbit tersimpan dalam *database*.	Data penerbit berhasil disimpan dalam *database*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Tambah penerbit			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data kategori contoh : Kategori : Gramedia	Penerbit tidak dapat disimpan dalam *database* karena data penerbit sudah ada dan muncul pesan kesalahan	Sistem tidak dapat menyimpan data penerbit dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

12. Pengujian ubah penerbit

Pengujian ubah penerbit dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian ubah penerbit dilihat pada tabel 4.41

Tabel 4.41 Pengujian ubah penerbit

Kasus dan Hasil Uji Benar (Data Benar)			
Ubah penerbit			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Ubah penerbit	Menampilkan form ubah penerbit beserta data yang akan diubah	Dapat menampilkan form ubah penerbit beserta data yang akan diubah	[√] Diterima [ ] Ditolak
Data penerbit lama contoh : Penerbit : Gramedia	Tercantun dalam *field* penerbit berupa nama penerbit.	Dapat menampilkan field data penerbit yang akan diubah.	[√] Diterima [ ] Ditolak
Data penerbit yang akan diubah contoh : Kategori : Graha ilmu	Dapat mengisi *field* data yang akan diubah dengan benar dan data penerbit berhasil diubah dan disimpan dalam *database*.	Dapat mengisi *field* data yang akan diubah dengan benar dan data penerbit berhasil diubah dan disimpan dalam *databsase*	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji Salah (Data Salah)			
Ubah penerbit			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data penerbit yang akan diubah contoh : penerbit: Penerbit Gramedia graha ilmu dan segala jenis penerbit	Data penerbit tidak berhasil diubah dan menampilkan pesan kesalahan, maxlength adalah 32	Sistem tidak dapat menyimpan data penerbit dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

13. Pengujian hapus penerbit

Pengujian hapus penerbit dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian hapus penerbit dilihat pada tabel 4.42

Tabel 4.42 Pengujian hapus penerbit

Kasus dan Hasil Uji Benar (Data Benar)			
Hapus penerbit			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data penerbit yang akan dihapus : contoh : penerbit: Gramedia	Tercantum data penerbit yang akan dihapus berupa setelah memilih hapus maka tampil pesan peringatan apakah data akan dihapus	Tercantum data penerbit yang akan dihapus berupa setelah memilih hapus maka tampil pesan peringatan apakah data akan dihapus	[√] Diterima [ ] Ditolak

14. Pengujian cari penerbit

Pengujian cari penerbit dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian cari penerbit dilihat pada tabel 4.43

Tabel 4.43 Pengujian cari penerbit

Kasus dan Hasil Uji Benar (Data Benar)			
Cari penerbit			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data penerbit	Menampilkan data penerbit	Berhasil menampilkan data penerbit	[√] Diterima [ ] Ditolak

Data yang akan dicari berdasarkan nama penerbit contoh : Gramedia	Menampilkan data penerbit berdasarkan nama penerbit yang dicari	Berhasil Menampilkan data penerbit berdasarkan nama penerbit yang dicari	[√] Diterima [ ] Ditolak

15. Pengujian tambah buku

Pengujian tambah buku dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian tambah buku dilihat pada tabel 4.44

Tabel 4.44 Pengujian tambah buku

Kasus dan Hasil Uji Benar (Data Benar)			
Tambah buku			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Tambah buku	Menampilkan *form* tambah buku	Sistem berhasil menampilkan *form* tambah buku	[√] Diterima [ ] Ditolak
Mengisi data buku contoh : Kategori : Agama Subkategori : Agama Islam judul : Panduan Shalat Lengkap Sunnah Dan Wajib isbn : 9789791926997 pengarang : Sahid Sulaiman deskripsi : - Kuantitas : 10 Harga : 10000 Berat : 0.5 Ukurang : 12x13x14 gambar : img.jpg Penerbit : Gramedia	Tercantun dalam masing-masing *field* kemudian data buku tersimpan dalam *database*.	Data buku berhasil disimpan dalam *database*	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji Salah (Data Salah)			
Tambah buku			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data buku contoh : Kategori : Agama Subkategori : Agama Islam judul : Panduan Shalat Lengkap Sunnah Dan Wajib isbn : 9789791926997 pengarang : Sahid Sulaiman deskripsi : - Kuantitas : 10 Harga : 10000 Berat : 0.5 Ukurang : 12x13x14 gambar : img.jpg Penerbit : Gramedia	Data buku tidak dapat disimpan dalam *database* karena data buku sudah ada dan muncul pesan kesalahan	Sistem tidak dapat menyimpan data buku dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

16. Pengujian ubah buku

Pengujian ubah buku dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian ubah buku dilihat pada tabel 4.45

Tabel 4.45 Pengujian ubah buku

Kasus dan Hasil Uji Benar (Data Benar)			
Ubah buku			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Ubah buku	Menampilkan form ubah buku beserta data yang akan diubah	Dapat menampilkan form ubah buku beserta data yang akan diubah	[√] Diterima [ ] Ditolak
Data buku lama contoh : Kategori : Agama Subkategori : Agama Islam judul : Panduan Shalat Lengkap Sunnah Dan Wajib isbn : 9789791926997 pengarang : Sahid Sulaiman deskripsi : - Kuantitas : 10 Harga : 10000 Berat : 0.5 Ukurang : 12x13x14 gambar : img.jpg Penerbit : Gramedia	Tercantun dalam *field* buku berupa nama kategori, subkategori, judul, isbn, pengarang, deskripsi, harga, berat, gambar, ukuran, dan penerbit.	Dapat menampilkan field data buku yang akan diubah.	[√] Diterima [ ] Ditolak

Data buku yang akan diubah contoh : Kategori : Agama Subkategori : Agama Islam judul : Tuntunan Shalat 5 waktu isbn : 9789791926997 pengarang : Sahid Sulaiman deskripsi : - Kuantitas : 10 Harga : 10000 Berat : 0.5 Ukurang : 12x13x14 gambar : img.jpg Penerbit : Gramedia	Dapat mengisi *field* data yang akan diubah dengan benar dan data buku berhasil diubah dan disimpan dalam *database*.	Dapat mengisi *field* data yang akan diubah dengan benar dan data buku berhasil diubah dan disimpan dalam *databsase*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Ubah buku			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data buku yang akan diubah contoh : Kategori : Agama Subkategori : Agama Islam judul : Tuntunan Shalat 5 waktu isbn : 978979192699788273 pengarang : Sahid Sulaiman deskripsi : - Kuantitas : 10 Harga : 10000 Berat : 0.5 Ukurang : 12x13x14 gambar : img.jpg Penerbit : Gramedia	Data buku tidak berhasil diubah dan menampilkan pesan kesalahan, maxlength isbn adalah 32	Sistem tidak dapat menyimpan data buku dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

17. Pengujian hapus buku

Pengujian ubah buku dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian hapus buku dilihat pada tabel 4.46

Tabel 4.46 Pengujian hapus buku

Kasus dan Hasil Uji Benar (Data Benar)			
Hapus buku			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data buku yang akan dihapus contoh : Kategori : Agama Subkategori : Agama Islam judul : Tuntunan Shalat 5 waktu isbn : 978979192699788273 pengarang : Sahid Sulaiman deskripsi : - Kuantitas : 10 Harga : 10000 Berat : 0.5 Ukurang : 12x13x14 gambar : img.jpg Penerbit : Gramedia	Tercantum data buku yang akan dihapus berupa setelah memilih hapus maka tampil pesan peringatan apakah data akan dihapus	Tercantum data buku yang akan dihapus berupa setelah memilih hapus maka tampil pesan peringatan apakah data akan dihapus	[√] Diterima [ ] Ditolak

18. Pengujian cari buku

Pengujian cari buku dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian cari buku dilihat pada tabel 4.47

Tabel 4.47 Pengujian cari buku

Kasus dan Hasil Uji Benar (Data Benar)			
Cari Buku			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data buku	Menampilkan data buku	Berhasil menampilkan data buku	[√] Diterima [ ] Ditolak
Data yang akan dicari berdasarkan nama buku contoh : Shalat	Menampilkan data buku berdasarkan nama buku yang dicari	Berhasil Menampilkan data buku berdasarkan nama buku yang dicari	[√] Diterima [ ] Ditolak

19. Pengujian pengadaan

Pengujian pengadaan dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian pengadaan buku dilihat pada tabel 4.48

Tabel 4.48 Pengujian pengadaan buku

Kasus dan Hasil Uji Benar (Data Benar)			
Pengadaan Buku			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pengadaan stok buku	Menampilkan *form* pengadaan stok buku	Berhasil menampilkan form pengadaan stok buku	[√] Diterima [ ] Ditolak
Mengisi jumlah pemesanan buku contoh : 10	Tercantun dalam *field* jumlah pemesanan dan data pengadaan buku tersimpan dalam *database*	Data pengadaan buku berhasil disimpan dalam *database*	[√] Diterima [ ] Ditolak
Cetak Pemesanan	Menampilkan surat pemesanan buku dalam bentuk pdf	Sistem menampilkan surat pemesanan buku dalam bentuk pdf	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Pengadaan buku			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi jumlah pemesanan buku contoh : ABC	Data pengadaan buku tidak berhasil diubah dan menampilkan pesan kesalahan, tipe data pemesanan adalah *integer*	Sistem tidak berhasil menyimpan data pengadaan stok buku	[√] Diterima [ ] Ditolak

20. Pengujian pengolahan data pemesanan

Pengujian pengolahan data pemesanan dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian pengolahan data pemesanan dapat dilihat pada tabel 4.49

Tabel 4.49 Pengujian pengolahan data pemesanan

Kasus dan Hasil Uji Benar (Data Benar)			
Pengolahan data pemesanan			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih status pemesanan yang akan ditampilkan	Tampil data pemesanan sesuai dengan status pemesanan.	Sistem berhasil menampilkan data pemesanan	[√] Diterima [ ] Ditolak

		berdasarkan status pemesanan	
Pilih status pemesanan tunggu konfirmasi	Tampil data pemesanan sesuai dengan status pemesanan tunggu konfirmasi.	Sistem berhasil menampilkan data pemesanan tunggu konfirmasi	[√] Diterima [ ] Ditolak
Pilih detail pemesanan pada data pemesanan dengan status tunngu konfirmasi	Tampil detail pemesanan dengan data pemesanan tunggu konfirmasi	Sistem berhasil menampilkan data pemesanan dengan status pemesanan tunggu konfirmasi	[√] Diterima [ ] Ditolak
Pilih status tidak valid	Menampilkan text area untuk mengisi alasan tidak valid pada pemesanan	Sistem berhasil menampilkan text area ketika pilih status tidak valid	[√] Diterima [ ] Ditolak
Pilih simpan	Menyimpan pesan pada pemesanan dan mengubah status pemesanan menjadi tidak valid	Sistem berhasil menyimpan pesan pada pemesanan dan mengubah status pemesanan menjadi tidak valid	[√] Diterima [ ] Ditolak
Pilih cek lunas	Menyimpan status pembayaran menjadi lunas	Sistem berhasil menyimpan status pembayaran menjadi lunas	[√] Diterima [ ] Ditolak
Pilih status pembayaran lunas	Menampilkan pemesanan dengan status pemesanan lunas	Sistem berhasil menampilkan pemesanan dengan status pemesanan lunas	[√] Diterima [ ] Ditolak
Pilih menu ubah	Menampilkan form ubah pemesanan	Sistem berhasil menampilkan form ubah pemesanan	[√] Diterima [ ] Ditolak
Memasukan status pengiriman dan nomor resi pengiriman	Jika salah satu field yang bertanda wajib diisi tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	Sistem dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak
Pilih tombol Simpan	Data pemesanan berhasil disimpan dengan status pengiriman terkirim beserta nomor resi pengiriman	Data berhasil menyimpan data dengan status pengiriman terkirim beserta nomor resi pengiriman	[√] Diterima [ ] Ditolak

21. Pengujian kirim Promosi

Pengujian kirim promosi dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian pengadaan kirim promosi dapat dilihat pada tabel 4.50

Tabel 4.50 Pengujian kirim promosi

Kasus dan Hasil Uji Benar (Data Benar)			
Pengujian kirim promosi			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih menu kirim promosi	Tampil form kirim promosi yang menampilkan data konsumen.	Sistem berhasil menampilkan data konsumen beserta.	[√] Diterima [ ] Ditolak
Memasukan gambar untuk dikirim ke konsumen	Gambar promosi dikirim kepada konsumen yang sudah 5 kali membeli	Sistem berhasil mengirim promosi kepada konsumen yang sudah 5 kali membeli	[√] Diterima [ ] Ditolak

22. Pengujian retur

Pengujian retur dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian pengadaan retur dapat dilihat pada tabel 4.51

Tabel 4.51 Pengujian retur

Kasus dan Hasil Uji Benar (Data Benar)			
Pengujian retur			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih menu retur	Tampil form yang berisi data retur	Sistem berhasil menampilkan data retur.	[√] Diterima [ ] Ditolak
Memilih proses retur	Memproses data retur	Sistem berhasil mengubah status data retur	[√] Diterima [ ] Ditolak

23. Pengujian balas kontak

Pengujian balas kontak dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian balas kontak dapat dilihat pada tabel 4.52

Tabel 4.52 Pengujian balas kontak

Kasus dan Hasil Uji Benar (Data Benar)			
Pengujian balas kontak			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih menu balas kontak	Memanpilkan data kontak yang sudah dikirimkan oleh konsumen	Sistem berhasil menampilkan data kontak yang dikirimkan oleh konsumen	[√] Diterima [ ] Ditolak
Memilih balas kontak	Menampilkan form balas kontak dan menampilkan pesan kesalahan apabila salah satu field tidak terisi	Sistem berhasil menampilkan form balas kontak dan menampilkan pesan kesalahan apabila salah satu field tidak terisi	[√] Diterima [ ] Ditolak

24. Pengujian tambah *user*

Pengujian tambah *user* dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian tambah *user* dilihat pada tabel 4.53

Tabel 4.53 Pengujian tambah user

Kasus dan Hasil Uji Benar (Data Benar)			
Tambah user			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Tambah *user*	Menampilkan *form* tambah *user*	Sistem berhasil menampilkan *form* tambah *user*	[√] Diterima [ ] Ditolak
Mengisi data *user* contoh : *username* : admin *password* : admin *level* : admin	Tercantun dalam masing-masing *field* kemudian data *user* tersimpan dalam *database*.	Data *user* berhasil disimpan dalam *database*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Tambah user			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data *user* contoh : *username* : admin *password* : admin level : -	Data *user* tidak dapat disimpan dalam *database* karena data level belum terisi	Sistem tidak dapat menyimpan data *user* dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

25. Pengujian ubah *user*

Pengujian ubah *user* dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian ubah *user* dilihat pada tabel 4.54

Tabel 4.54 Pengujian ubah *user*

Kasus dan Hasil Uji Benar (Data Benar)			
Ubah user			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Ubah *user*	Menampilkan form ubah *user* beserta data yang akan diubah	Dapat menampilkan form ubah *user* beserta data yang akan diubah	[√] Diterima [ ] Ditolak
Data *user* lama yang akan diubah contoh : *username* : admin *password* : admin *level* : admin	Tercantun dalam *field user* berupa *username*, *password*, dan *level*	Dapat menampilkan field data *user* yang akan diubah.	[√] Diterima [ ] Ditolak
Mengisi data *user* contoh : *username* : admin *password* : admin *level* : admin	Dapat mengisi *field* data yang akan diubah dengan benar dan data *user* berhasil diubah dan disimpan dalam *database*.	Dapat mengisi *field* data yang akan diubah dengan benar dan data *user* berhasil diubah dan disimpan dalam *databsase*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Ubah user			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data *user* yang akan diubah contoh : *username* : admin itu adalah administrator punya semua hak akses *password* : admin *level* : admin	Data *user* tidak berhasil diubah dan menampilkan pesan kesalahan, maxlength *username* adalah 20	Sistem tidak dapat menyimpan data *user* dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

26. Pengujian hapus *user*

Pengujian ubah *user* dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian hapus *user* dilihat pada tabel 4.55

Tabel 4.55 Pengujian hapus buku

Kasus dan Hasil Uji Benar (Data Benar)			
Hapus user			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data *user* yang akan dihapus contoh : *username* : admin *password* : admin *level* : admin	Tercantum data *user* yang akan dihapus berupa setelah memilih hapus maka tampil pesan peringatan apakah data akan dihapus	Tercantum data *user* yang akan dihapus berupa setelah memilih hapus maka tampil pesan peringatan apakah data akan dihapus	[√] Diterima [ ] Ditolak

27. Pengujian tambah grup *user*

Pengujian tambah grup *user* dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian tambah grup *user* dilihat pada tabel 4.56

Tabel 4.56 Pengujian tambah grup *user*

Kasus dan Hasil Uji Benar (Data Benar)			
Tambah grup user			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Tambah grup *user*	Menampilkan *form* tambah grup *user*	Sistem berhasil menampilkan *form* tambah grup *user*	[√] Diterima [ ] Ditolak
Mengisi data grup *user* contoh : nama grup : admin	Tercantun dalam masing-masing *field* kemudian data grup *user* tersimpan dalam *database*.	Data grup *user* berhasil disimpan dalam *database*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Tambah grup user			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data grup *user* contoh : nama grup : admin	Data *user* tidak dapat disimpan dalam *database* karena data nama grup sudah ada	Sistem tidak dapat menyimpan data grup *user* dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

28. Pengujian ubah grup *user*

Pengujian ubah grup *user* dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian ubah grup *user* dilihat pada tabel 4.57

Tabel 4.57 Pengujian ubah grup *user*

Kasus dan Hasil Uji Benar (Data Benar)			
Ubah grup user			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Ubah grup *user*	Menampilkan form ubah grup *user* beserta data yang akan diubah	Dapat menampilkan form ubah grup *user* beserta data yang akan diubah	[√] Diterima [ ] Ditolak
Data grup *user* lama yang akan diubah contoh : nama grup : admin	Tercantun dalam *field* grup *user* berupa nama grup	Dapat menampilkan field data grup *user* yang akan diubah.	[√] Diterima [ ] Ditolak
Mengisi data grup *user* contoh : nama grup : operator	Dapat mengisi *field* data yang akan diubah dengan benar dan data *user* berhasil diubah dan disimpan dalam *database*.	Dapat mengisi *field* data yang akan diubah dengan benar dan data *user* berhasil diubah dan disimpan dalam *databsase*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Ubah grup user			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data *user* yang akan diubah contoh : nama grup : admin itu adalah administrator punya semua hak akses	Data *user* tidak berhasil diubah dan menampilkan pesan kesalahan, maxlength *username* adalah 20	Sistem tidak dapat menyimpan data *user* dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

29. Pengujian cetak laporan penjualan

Pengujian cetak laporan penjualan dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian cetak laporan penjualan dilihat pada tabel 4.58

Tabel 4.58 Pengujian cetak laporan penjualan

Kasus dan Hasil Uji Benar (Data Benar)			
Ubah user			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Cetak laporan penjualan	Menampilkan form laporan penjualan	Dapat menampilkan form laporan penjualan	[√] Diterima [ ] Ditolak
Memasukan bulan dan tahun contoh :	Tercantun dalam *field* laporan berupa	Dapat menampilkan data laporan pada	[√] Diterima [ ] Ditolak

Bulan : Juni Tahun : 2011	bulan dan tahun	periode yang dipilih.	
Cetak laporan	Menampilkan laporan penjualan pada periode tertentu dalam bentuk *file* pdf.	Dapat menampilkan laporan penjualan pada periode tertentu dalam benuk *file* pdf	[√] Diterima [ ] Ditolak

30. Pengujian ubah akun

Pengujian ubah akun dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian ubah akun dilihat pada tabel 4.59

Tabel 4.59 Pengujian ubah akun

Kasus dan Hasil Uji Benar (Data Benar)			
Ubah user			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Ubah akun	Menampilkan form ubah akun beserta data yang akan diubah	Dapat menampilkan form ubah akun beserta data yang akan diubah	[√] Diterima [ ] Ditolak
Data akun lama yang akan diubah contoh : *username* : admin *password* : admin	Tercantun dalam *field* akun berupa *username*, *password*	Dapat menampilkan field data akun yang akan diubah.	[√] Diterima [ ] Ditolak
Mengisi data akun contoh : *username* : admin1 *password* : admin1	Dapat mengisi *field* data yang akan diubah dengan benar dan data akun berhasil diubah dan disimpan dalam *database*.	Dapat mengisi *field* data yang akan diubah dengan benar dan data akun berhasil diubah dan disimpan dalam *databsase*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Ubah user			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data akun yang akan diubah contoh : *username* : admin itu adalah administrator punya semua hak akses *password* : admin *level* : admin	Data akun tidak berhasil diubah dan menampilkan pesan kesalahan, maxlength *username* adalah 20	Sistem tidak dapat menyimpan data akun dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

4.6.2.2 Pengujian aplikasi konsumen

Pengujian yang dilakukan untuk halaman konsumen adalah menguji semua proses yang ada di halaman konsumen diantaranya adalah sebagai berikut :

1. Pengujian tambah barang ke keranjang

   Pengujian keranjang belanja dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian keranjang belanja dilihat pada tabel 4.60

Tabel 4.60 Pengujian tambah barang ke keranjang

Kasus dan Hasil Uji Benar (Data Benar)			
Tambah barang ke keranjang			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Keranjang belanja	Menampilkan halaman keranjang belanja yang berisi buku yang telah dibeli	Sistem Menampilkan halaman keranjang belanja yang berisi buku yang telah dibeli	[√] Diterima [ ] Ditolak
Mengisi data buku kategori contoh : buku : belajar agama islam	Buku masuk kedalam keranjang belanja.	Sistem berhasil memasukan data buku kedalam keranjang belanja	[√] Diterima [ ] Ditolak

2. Pengujian *update* keranjang

   Pengujian *update* keranjang dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian *update* keranjang dilihat pada tabel 4.61

Tabel 4.61 Pengujian *update* keranjang

Kasus dan Hasil Uji Benar (Data Benar)			
Update keranjang			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Keranjang belanja	Menampilkan halaman keranjang belanja yang berisi buku yang telah dibeli	Sistem Menampilkan halaman keranjang belanja yang berisi buku yang telah dibeli	[√] Diterima [ ] Ditolak

Data keranjang lama contoh : Buku : Belajar Agama Islam Jumlah 1	Tercantun dalam *field* keranjang berupa judul buku, jumlah.	Dapat menampilkan field data keranjang yang akan diubah.	[√] Diterima [ ] Ditolak
Data keranjang yang akan diubah contoh : Kategori : Belajar Agama Islam Jumlah : 100	Dapat mengisi *field* data yang akan diubah dengan benar dan data keranjang berhasil diubah dan apabila jumlah lebih dari stok maka akan muncul pesan akan memesan.	Dapat mengisi *field* data yang akan diubah dengan benar dan data keranjang berhasil diubah dan disimpan dalam *databsase*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Update keranjang			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data kategori yang akan diubah contoh : Judul : Belajar Agama Islam Jumlah : a	Data keranjang tidak berhasil diubah dan menampilkan pesan kesalahan, jumlah mempunyai tipe data integer	Sistem tidak dapat menyimpan data kategori dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

3. Pengujian hapus data barang dari keranjang

Pengujian hapus data barang dari keranjang dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian hapus kategori dilihat pada tabel 4.62

Tabel 4.62 Pengujian hapus data barang dari keranjang

Kasus dan Hasil Uji Benar (Data Benar)			
Hapus data barang dari keranjang			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data buku dalam keranjang yang akan dihapus : contoh : Judul : Belajar Agama Islam Jumlah : 1	Tercantum data buku dalam keranjang yang akan dihapus berupa setelah memilih hapus maka tampil pesan peringatan apakah data akan dihapus	Tercantum data buku dalam keranjang yang akan dihapus berupa setelah memilih hapus maka tampil pesan peringatan apakah data akan dihapus	[√] Diterima [ ] Ditolak

4. Pengujian konfirmasi pembayaran

Pengujian konfirmasi pembayaran dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian konfirmasi pembayaran dilihat pada tabel 4.63

Tabel 4.63 Pengujian konfirmasi pembayaran

Kasus dan Hasil Uji Benar (Data Benar)			
Konfirmasi pembayaran			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Konfirmasi pembayaran	Menampilkan *form* konfirmasi pembayaran	Sistem menampilkan *form* konfirmasi pembayaran	[√] Diterima [ ] Ditolak
Mengisi data konfirmasi pembayaran contoh : nomor faktur : 10 Tanggal transfer : 29-06-2011 Atas nama : samsul	Tercantun dalam masing-masing *field* kemudian data konfirmasi dan tersimpan dalam *database*.	Data konfirmasi pembayaran berhasil disimpan dalam *database*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Konfirmasi pembayaran			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data konfirmasi contoh : nomor faktur : 10 Tanggal transfer : 99-99-2011 Atas nama : samsul	data konfirmasi pembayaran tidak dapat disimpan dalam *database* karena format tanggal salah	Sistem tidak dapat menyimpan data konfirmasi pembayaran dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

5. Pengujian kontak kami

Pengujian kontak kami dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian kontak kami dilihat pada tabel 4.64

Tabel 4.64 Pengujian kontak kami

Kasus dan Hasil Uji Benar (Data Benar)			
Kontak kami			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kontak kami	Menampilkan *form* kontak kami	Sistem menampilkan *form*	[√] Diterima [ ] Ditolak

		kontak kami	
Mengisi data kontak kamu contoh : nama : samsul email : samsul@yahoo.com alamat : bandung pesan : hai hai hai telepon : 92429046	Tercantun dalam masing-masing *field* kemudian data kontak dan tersimpan dalam *database*.	Data konfirmasi kontak berhasil disimpan dalam *database*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Kontak kami			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data kontak contoh : nama : samsul email : samsul.yahoo.com alamat : bandung pesan : hai hai hai telepon : 92429046	data kontak tidak dapat disimpan dalam *database* karena format *email* salah	Sistem tidak dapat menyimpan data kontak pembayaran dalam ke dalam *database* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak

6. Pengujian cari buku

Pengujian cari buku dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan. Pengujian cari buku dilihat pada tabel 4.65

Tabel 4.65 Pengujian cari buku

Kasus dan Hasil Uji Benar (Data Benar)			
Cari Buku			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data buku	Menampilkan data buku	Berhasil menampilkan data buku	[√] Diterima [ ] Ditolak
Data yang akan dicari berdasarkan nama buku contoh : Agama	Menampilkan data buku berdasarkan buku buku yang dicari	Berhasil Menampilkan data buku berdasarkan nama buku yang dicari	[√] Diterima [ ] Ditolak

4.6.3 Kesimpulan Hasil Pengujian *Alpha*

Berdasarkan hasil pengujian sistem yang telah dilakukan secara keseluruhan maka didapat kesimpulan bahwa proses pada situs *e- commerce* pada

toko Rumah Buku secara fungsional sistem sudah dapat digunakan dan menghasilkan output yang diharapkan.

4.6.4 Pengujian *Betha*

Pengujian *betha* merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan, dengan menggunakan kuesioner mengenai kepuasan user terhadap aplikasi yang telah dibangun, dengan kandungan poin syarat *user friendly* serta apakah sesuai dengan yang diharapkan. Adapun metode penilaian pengujian yang digunakan yaitu metode kuantitatif berdasarkan data sampel dari *user*.

Kuesioner disebar menggunakan teknik sampling, yaitu dengan menyebarkan kuesioner pegawai toko buku dan konsumen toko buku. Adapun alasan penggunaan teknik sampling dikarenakan populasi dari penelitian ini diketahui. Prasyarat dari responden penelitian ini adalah :

1. Konsumen yang ada pada Toko Rumah Buku Bandung
2. *User* dari pegawai toko yang berpotensi sebagai admin dan petugas

Adapun beberapa poin yang diteliti adalah :

1. *Website* dapat memenuhi kebutuhan informasi pelanggan.
2. Tampilan antar muka yang *user friendly*.
3. Memudahkan pelanggan dalam melakukan transaksi pembelian barang.
4. Pendokumentasikan data transaksi penjualan akan lebih teratur bagi Toko Rumah Buku Bandung
5. Sistem pembayaran yang sangat mudah digunakan untuk pembayaran. *Website* menjadi sarana promosi untuk memperluas cakupan pasar

Berdasarkan data hasil kusioner tersebut, dapat dicari prosentase masing-masing jawaban dengan menggunakan rumus :

Keterangan :

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

Y = Nilai prosentase

Berikut ini adalah hasil persentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas. Kuisioner ini diujikan kepada 10 orang.

4.6.4.1 Kuesioner konsumen

Kuesioner untuk konsumen terdiri dari 5 pertanyaan (contoh kuesioner dapat diliihat pada lampiran) yang disebarkan kepada sepuluh orang konsumen dan pengunjung untuk mengetahui kepuasan konsumen dengan menggunakan skala likert dengan skala 1 sampai 7, dengan ketentuan skala untuk tiap pertanyaan sebagai berikut :

1. Menurut anda, apakah anda setuju bahwa *website e-commerce* pada toko Rumah Buku akan lebih memudahkan konsumen dalam melakukan transaksi pembelian buku?

Tabel 4.66 Kuisioner pertanyaan nomor 1 konsumen

Pertanyaan	No	Keterangan	Responden	Prosentase (%)
1	1	Sangat Setuju		
	2	Setuju	7	70
	3	Cukup Setuju	3	30
	4	Biasa Saja		

	5	Kurang Setuju		
	6	Tidak Setuju		
	7	Sangat Tidak Setuju		
Jumlah			10	100

Berdasarkan hasil prosentase pada tabel 4.66, maka dapat disimpulkan sebanyak 7 orang atau 70 % menyatakan setuju, 3 orang atau 30% menyatakan cukup setuju bahwa akan lebih memudahkan pelanggan dalam melakukan transaksi pembelian barang.

2. Menurut anda, apakah anda setuju bahwa informasi barang yang diberikan oleh *website e-commerce* Rumah buku sudah sesuai dengan informasi yang anda butuhkan?

Tabel 4.67 Kuisioner pertanyaan nomor 2 konsumen

Pertanyaan	No	Keterangan	Responden	Prosentase (%)
2	1	Sangat Setuju		
	2	Setuju	8	80
	3	Cukup Setuju	2	20
	4	Biasa Saja		
	5	Kurang Setuju		
	6	Tidak Setuju		
	7	Sangat Tidak Setuju		
Jumlah			10	100

Berdasarkan hasil prosentase pada tabel 4.67, maka dapat disimpulkan sebanyak 8 atau 80% menyatakan setuju, dan 2 atau 20% menyatakan cukup setuju bahwa informasi barang yang diberikan sudah sesuai dengan informasi yang dibutuhkan.

3. Menurut anda, apakah anda setuju dengan adanya *website e-commerce* Rumah Buku, pihak Toko Rumah Buku Bandung akan lebih mudah mengelola data *master* dan data transaksi penjualan?

Tabel 4.68 Kuisioner pertanyaan nomor 3 konsumen

Pertanyaan	No	Keterangan	Responden	Prosentase (%)
3	1	Sangat Setuju		
	2	Setuju	8	80
	3	Cukup Setuju	2	20
	4	Biasa Saja		
	5	Kurang Setuju		
	6	Tidak Setuju		
	7	Sangat Tidak Setuju		
Jumlah			10	100

Berdasarkan hasil prosentase pada tabel 4.68, maka dapat disimpulkan sebanyak 8 orang atau 80 % menyatakan setuju, 2 orang atau 30% menyatakan cukup setuju dengan adanya website *e-commerce* pada Rumah Buku akan lebih mempermudah mengelola data *master* dan data transaksi penjualan.

4. Menurut anda, apakah anda setuju bahwa sistem pembayaran yang digunakan akan lebih memudahkan dalam proses transaksi?

Tabel 4.69 Kuisioner pertanyaan nomor 4 konsumen

Pertanyaan	No	Keterangan	Responden	Prosentase (%)
4	1	Sangat Setuju		
	2	Setuju	7	70
	3	Cukup Setuju	3	30
	4	Biasa Saja		
	5	Kurang Setuju		
	6	Tidak Setuju		
	7	Sangat Tidak Setuju		
Jumlah			10	100

Berdasarkan hasil prosentase pada tabel 4.69, maka dapat disimpulkan sebanyak 7 orang atau 70 % menyatakan setuju, 3 orang atau 30% menyatakan cukup setuju bahwa sistem pembayaran yang digunakan akan lebih memudahkan dalam proses transaksi.

5. Menurut anda, apakah anda setuju bahwa tampilan *website* cukup menarik dan *user friendly*?

Tabel 4.70 Kuisioner pertanyaan nomor 5 konsumen

Pertanyaan	No	Keterangan	Responden	Prosentase (%)
5	1	Sangat Setuju	4	40
	2	Setuju	6	60
	3	Cukup Setuju		
	4	Biasa Saja		
	5	Kurang Setuju		
	6	Tidak Setuju		
	7	Sangat Tidak Setuju		
Jumlah			10	100

Berdasarkan hasil prosentase pada tabel 4.70, maka dapat disimpulkan sebanyak 4 orang atau 40 % menyatakan sangat setuju, 6 orang atau 60% menyatakan setuju bahwa tampilan dari website *e-commerce* pada toko Rumah Buku menarik dan *user friendly*.

4.6.4.2 Kuesioner *admin*

Kuesioner untuk *admin* terdiri dari 5 pertanyaan (contoh kuesioner dapat diliihat pada lampiran) yang disebarkan kepada 2 orang pengguna dengan menggunakan skala likert dengan skala 1 sampai 7, dengan ketentuan skala untuk tiap pertanyaan sebagai berikut :

1. Menurut anda, apakah *website e-commerce* Rumah Buku yang dibangun dapat membantu meningkatkan penjualan buku ?

Tabel 4.71 Kuisioner pertanyaan nomor 1 admin

Pertanyaan	No	Keterangan	Responden	Prosentase (%)
1	1	Sangat membantu		
	2	Membantu	1	50
	3	Cukup membantu	1	50
	4	Biasa Saja		

	5	Kurang membantu		
	6	Tidak membantu		
	7	Sangat Tidak membantu		
Jumlah			2	100

Berdasarkan hasil prosentase pada tabel 4.71, telah dihitung jawaban kuesioner dengan menggunakan rumus 1 maka dapat disimpulkan bahwa sebanyak 1 orang atau 50% mengatakan bahwa perangkat lunak yang dibuat dapat membantu meningkatkan penjualan produk dan 1 orang atau 50% mengatakan perangkat lunak tersebut cukup membantu meningkatkan penjualan produk sehingga secara keseluruhan mengatakan keseluruhan *website e-commerce* tersebut cukup membantu.

2. Menurut anda, apakah *website e-commerce* Rumah Buku yang dibangun dapat memudahkan dalam mengelola data buku?

Tabel 4.72 Kuisioner pertanyaan nomor 2 admin

Pertanyaan	No	Keterangan	Responden	Prosentase (%)
2	1	Sangat memudahkan		
	2	Memudahkan	1	50
	3	Cukup memudahkan	1	50
	4	Biasa Saja		
	5	Kurang memudahkan		
	6	Tidak memudahkan		
	7	Sangat Tidak memudahkan		
Jumlah			2	100

Berdasarkan hasil prosentase pada tabel 4.72, telah dihitung jawaban kuesioner dengan menggunakan rumus 1 maka dapat disimpulkan bahwa sebanyak 1 orang atau 50% mengatakan bahwa perangkat lunak yang dibuat memudahkan dalam mengelola data produk dan 1 orang atau 50% mengatakan perangkat lunak tersebut cukup memudahkan dalam mengelola

data produk sehingga secara keseluruhan *website e-commerce* tersebut cukup memudahkan.

3. Menurut anda, apakah *website e-commerce* Rumah Buku yang dibangun dapat memudahkan dalam membuat laporan penjualan buku pada toko Rumah Buku?

Tabel 4.73 Kuisioner pertanyaan nomor 3 admin

Pertanyaan	No	Keterangan	Responden	Prosentase (%)
3	1	Sangat memudahkan		
	2	Memudahkan		
	3	Cukup memudahkan	2	100
	4	Biasa Saja		
	5	Kurang memudahkan		
	6	Tidak memudahkan		
	7	Sangat Tidak memudahkan		
Jumlah			2	100

Berdasarkan hasil prosentase pada tabel 4.73, telah dihitung jawaban kuesioner dengan menggunakan rumus 1 maka dapat disimpulkan bahwa sebanyak 2 orang atau 100% mengatakan bahwa *website e-commerce* yang dibangun cukup memudah kan dalam pembuatan laporan penjualan.

4. Menurut anda, apakah *website e-commerce* Rumah Buku yang dibangun dapat memperluas pemasaran penjualan ?

Tabel 4.74 Kuisioner pertanyaan nomor 4 admin

Pertanyaan	No	Keterangan	Responden	Prosentase (%)
4	1	Sangat memperluas		
	2	Memperluas	1	50
	3	Cukup memperluas	1	50
	4	Biasa Saja		
	5	Kurang memperluas		
	6	Tidak memperluas		
	7	Sangat Tidak memperluas		
Jumlah			2	100

Berdasarkan hasil prosentase pada tabel 4.74, telah dihitung jawaban kuesioner dengan menggunakan rumus 1 maka dapat disimpulkan bahwa sebanyak 1 orang atau 50% mengatakan bahwa perangkat lunak yang dibuat memperluas pemasaran penjualan produk dan 1 orang atau 50% mengatakan keseluruhan *website e-commerce* tersebut dibuat cukup memperluas pemasaran penjualan produk sehingga secara keseluruhan perangkat lunak tersebut cukup memperluas.

5. Menurut anda, apakah dengan adanya *website e-commerce* Rumah Buku dapat membantu dalam sarana promosi dan memperkenalkan toko Rumah Buku kepada masyarakat luas ?

Tabel 4.75 Kuisioner pertanyaan nomor 5 admin

Pertanyaan	No	Keterangan	Responden	Prosentase (%)
5	1	Sangat membantu		
	2	Membantu	1	50
	3	Cukup membantu	1	50
	4	Biasa Saja		
	5	Kurang membantu		
	6	Tidak membantu		
	7	Sangat tidak membantu		
Jumlah			2	100

Berdasarkan hasil prosentase pada tabel 4.75, telah dihitung dengan menggunakan rumus 1 maka dapat disimpulkan bahwa sebanyak 1 orang atau 50% mengatakan bahwa *website e-commerce* yang dibuat sangat membantu dalam sarana promosi dan 1 orang atau 50% mengatakan perangkat lunak tersebut membantu dalam sarana promosi sehingga secara keseluruhan *website e-commerce* tersebut membantu.

4.6.5 Kesimpulan Pengujian *Betha*

Berdasarkan pengujian *betha* yang ada pada subbab sebelumnya, dapat diambil kesimpulan bahwa:

1. Aplikasi *E-Commerce* berbasis *web* di Toko Rumah Buku Bandung memudahkan pelanggan dalam mendapatkan informasi barang yang tersedia.
2. Konsumen dapat melakukan transaksi pembelian barang di Toko Rumah Buku Bandung dengan lebih mudah.
3. Proses pembayaran yang digunakan cukup mudah digunakan bagi para pelanggan.
4. Aplikasi yang dibangun dapat memudahkan pihak Toko Rumah Buku Bandung dalam mendokumentasikan data *master* dan data transaksi penjualan barang.